Journal of Computer Applications
计算机应用,2012,32(11):3067-3070
ISSN 1001-9081
CODEN JYIIDU
2012-11-01
http://www.joca.cn

文章编号:1001-9081(2012)11-3067-04
doi:10.3724/SP.J.1087.2012.03067

网构软件的随机性资源自适应性的形式化分析与验证

夏　琦,王忠群*
(安徽工程大学,计算机与信息学院,安徽 芜湖 241000)
(*通信作者电子邮箱 zqwang@ahpu.edu.cn)

摘　要:因特网上的资源具有不确定性、随机性,需要考虑如何保证网构软件系统在运行中满足资源需求。使用随机性资源接口自动机对软件构件的行为进行形式化建模,并使用随机性资源接口自动机网络描述构件组装系统的组合行为;在资源不确定的情况下,检验组合系统是否满足资源约束,并提出基于可达图的相应算法。给出了一个实例网上书店系统,并用模型检测工具 Spin 验证了模型的正确性。

关键词:网构软件;随机性资源;接口自动机;可达图
中图分类号:TP311.5　　**文献标志码**:A

Formal analysis and verification of randomness resources for Internetware

XIA Qi, WANG Zhong-qun*
(*School of Computer and Information, Anhui Polytechnic University, Wuhu Anhui* 241000, *China*)

Abstract: The resources on Internet are uncertain and random, which poses a challenge on how to guarantee that requirements of the resources are met for an Internetware system at run time. The interface automata with randomness resources was used to model the behaviors of software component, and the combination behavior of component assembly system was described using the randomness resources interface automata networks. Under the circumstance of resources with uncertainty and randomness, whether all the behaviors of a system were satisfied within the specified resource constraints was checked. And a reached graph based algorithm was proposed. Finally, the online bookstore system was used to illustrate the work, and the model checker Spin was used to verify the correctness of the proposed approach.

Key words: Internetware; randomness resources; interface automata; reached graph

0　引言

针对因特网的不确定环境[1-2],如何对因环境变化所导致的构件组合系统的行为变化进行有效的分析和验证,以提高网构软件系统的可信度给我们提出了挑战。文献[3]利用构件消息自动机对构件交互模型进行建模以验证构件行为可信性;文献[4]介绍了一种形式化的组件化软件过程建模方法;文献[5]使用带资源语义信息的接口自动机对软件构件的行为进行形式化建模及分析。在默认资源条件满足的前提下,研究了从验证业务一致性的角度来提高网构软件演化的可信性[6]。上述文献都很少考虑环境资源的随机性问题,而资源随机性恰是因特网的典型特征,因此,研究资源不确定条件下网构软件的可信性具有重要的意义。

本文首先扩展了接口自动机使其能够描述随机性资源语义信息,称其为随机性资源接口自动机,然后应用它来描述网构软件构件的组合行为,研究系统的行为是否满足随机性的资源约束,并给出检测随机性资源的可满足性、最小资源需求量算法。最后,使用模型检测工具 Spin 验证模型的正确性。

1　随机性资源接口自动机

接口自动机[7]用于描述软件构件接口的时序行为。对其进行随机性资源语义的扩展,将其称为随机性资源接口自动机。首先对系统资源使用进行以下假设。

1)资源是非共享的,即一种资源只能被一个构件使用,不能同时被多个构件使用。

2)资源是可以量化的,即为最小单位的整数倍。为了具有针对性,本文以网络带宽资源为例并进行量化处理,量化后的带宽资源最小单位为 256 kBps。

3)资源的随机性服从两点分布。

因为受到网络带宽资源的约束,现实应用中,如网上书店和电子邮箱等系统,一般会提供多个版本的系统,如文本版和标准版。当带宽资源不足时,用户登录文本版的系统以满足响应速度,有时甚至不能使用系统。针对网上书店系统[6],下面使用随机性资源接口自动机对其进行部分建模说明。图 1 为网上书店系统连接件的角色 seller 对应的接口自动机模型的扩充随机性资源语义,描述了系统对网络带宽资源的使用信息,每个状态都标有一个带概率信息的有序数〈(a_1,p),(a_2,1-p)〉(a_1、a_2 为正整数,$0 \leqslant p \leqslant 1$),该有序数表示系统对网络带宽资源的使用情况:占用 a_1 个单位的概率为 p,占用 a_2 个单位的概率为 1-p。初始状态 0,资源使用特征为〈(1,0.3),(2,0.7)〉,即构件在 0 状态下使用 1 个单位网络带宽资源的概率为 0.3,使用 2 个单位网络带宽资源的概率为 0.7;构件通过输出动作 goods 的刺激转换为状态 1,状态 1 的资源使用特征为〈(2,0.4),(3,0.6)〉,表示构件在此状态下

收稿日期:2012-05-28;**修回日期**:2012-07-18。
基金项目:国家自然科学基金资助项目(71171002);安徽省自然科学基金资助项目(070412058)。
作者简介:夏琦(1988-),男,安徽马鞍山人,硕士研究生,主要研究方向:软件工程;　王忠群(1965-),男,安徽芜湖人,教授,主要研究方向:软件工程、分布式计算、工作流技术。

使用2个单位网络带宽资源的概率为0.4,使用3个单位网络带宽资源的概率为0.6。角色 seller 的资源使用行为可以以此类推。这样,通过对每个状态上网络带宽资源使用情况的分析,可得知构件是否能够在网络带宽不稳定的情况下正常运行。因此,随机性资源接口自动机模型能够描述网上书店系统在网络带宽不稳定情况下的行为。

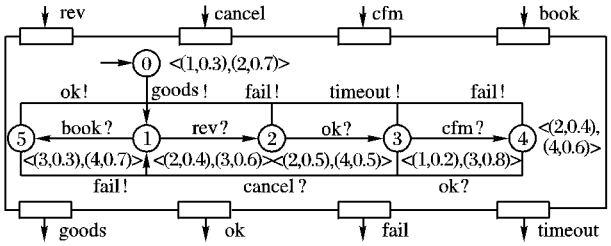


图1 seller 的随机性资源接口自动机

1.1 随机性资源接口自动机的形式化定义

随机性资源接口自动机形式化定义如下。

定义1 一个随机性资源接口自动机为 $P = \langle V_P, V_P^{\text{init}}, A_P^I, A_P^O, A_P^H, F_P, W_P, \Delta_P \rangle$,其中:

V_P 是 P 的状态集合。

$V_P^{\text{init}} \subseteq V_P$,是 P 的初始状态集合,其中最多包含一个状态。

A_P^I、A_P^O、A_P^H 是两两互不相交的动作集合,分别称为 P 的输入动作集合、输出动作集合、内部动作集合。所有的动作的集合记为 A_P,即 $A_P = A_P^I \cup A_P^O \cup A_P^H$。

F_P 是一个有穷映射集,$F_P = \{(f_p^1, \rho_p^1), (f_p^{1'}, 1-\rho_p^1); (f_p^2, \rho_p^2), (f_p^{2'}, 1-\rho_p^2); \cdots; (f_p^K, \rho_p^K), (f_p^{K'}, 1-\rho_p^K)\}$($K$为正整数,表示系统使用资源的种类数)。其中任意一个映射$(f_p^n, \rho_p^n), (f_p^{n'}, 1-\rho_p^n)$包含使用资源的数量和其概率,用 $F_P(v_i)$ 表示状态 v_i 上的资源信息。

W_P 是一个有穷集,每个元素形如:$w_i = \{(v_i, F_P(v_i)) \mid v_i \in V_P\}$。

Δ_P 是一个迁移关系,表示一组迁移部,$\Delta_P \subseteq W_P \times A_P \times W_P$。其中,$W_P$ 中的每一个 w_i 为一个带随机性资源约束的状态。$F_P(v_i)$ 用来表示状态 v_i 上的资源使用特征。

随机性资源接口自动机的行为定义与资源接口自动机[5]类似,其中状态转换序列 φ 表达了构件在转换状态的过程中,对资源使用的概率信息。图1中由状态0转换到状态1的随机性资源状态转换序列为 $\varphi = \{0, \langle (1,0.3), (2,0.7) \rangle\} \xrightarrow{\text{goods!}} \{1, \langle (2,0.4), (3,0.6) \rangle\}$。

1.2 随机性资源接口自动机网络

随机性资源接口自动机可以表示系统每个构件的资源使用行为,整个系统的资源使用行为使用多个随机性资源接口自动机的组合表示。这样组合的随机性资源接口自动机称为随机性资源接口自动机网络。随机性资源接口自动机网络在定义上类似于资源接口自动机网络,其定义详细内容见文献[5]。下面给出随机性资源接口自动机网络的状态集和动作集。

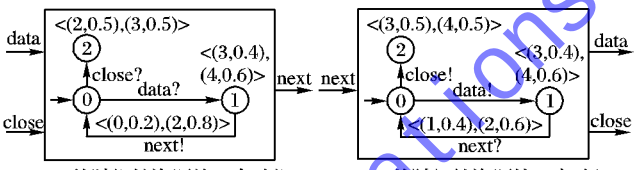
定义2 设 $N = (Q, Z)$,则 N 中的组合状态集和动作集定义如下:

N 中每个组合状态形如:$\bar{v} = (v_1, v_2, \cdots, v_n)$,其中 $v_i \in V_{P_i}$。其组合状态的集合为 $V_N = V_{P_1} \times V_{P_2} \times \cdots \times V_{P_n}$。

N 中的资源组合状态形如:$\bar{w} = (\bar{v}, F_N(\bar{v}))$,其中 $F_N(\bar{v})$ 是表示 N 的资源使用特征的有序集:$\langle (f_N^1(\bar{v}), \rho_N^1(\bar{v})), (f_N^{1'}(\bar{v}), 1-\rho_N^1(\bar{v})); (f_N^2(\bar{v}), \rho_N^2(\bar{v})), (f_N^{2'}(\bar{v}), 1-\rho_N^2(\bar{v})); \cdots; (f_N^K(\bar{v}), \rho_N^K(\bar{v})), (f_N^{K'}(\bar{v}), 1-\rho_N^K(\bar{v})) \rangle$。

N 动作集为 $A_N = A_N^I \cup A_N^O \cup A_N^H$,其中:输入动作为 $A_N^I = \frac{\left(\bigcup\limits_{1 \leqslant i \leqslant n} A_{P_i}^I\right)}{Z}$,输出动作为 $A_N^O = \frac{\left(\bigcup\limits_{1 \leqslant i \leqslant n} A_{P_i}^O\right)}{Z}$,内部动作集为 $A_N^H = \left(\bigcup\limits_{1 \leqslant i \leqslant n} A_{P_i}^I\right) \cup Z$。

如图2所示的随机性资源接口自动机 In、Out 分别描述数据输入端口和数据输出端口。In、Out 随机性资源接口自动机组合而成的随机性资源接口自动机网络可表示为(Q, Z)。其中:$Q = \{\text{In}, \text{Out}\}$,$Z = \{shared(\text{In}, \text{Out})\}$,$shared(\text{In}, \text{Out}) = (A_{In}^O \cap A_{Out}^I) \cup (A_{In}^I \cap A_{Out}^O) = \{\text{data}, \text{next}, \text{close}\}$。



(a) In的随机性资源接口自动机 (b) Out的随机性资源接口自动机

图2 随机性资源接口自动机网络

2 检验系统行为是否满足资源约束

构件组合系统在运行过程中所需要的资源必须是环境所能提供的。环境所能提供的资源最大量定义为 $L = \langle l_1, l_2, \cdots, l_K \rangle$,每一个 l_i 表示环境所能提供资源 ε_i 的使用数量。显然,最大资源量就构成构件组合系统的资源约束条件。

文献[8]指出一个兼容的接口自动机网络可通过 Petri-net 构造出与该自动机网络具有等价状态空间的可达图,故可知,由随机性资源接口自动机网络构造的可达图拥有该自动机网络的状态,可达图中的任意节点与相应组合状态具有一致的资源使用特征,用 $X_{s_j} = \langle (x_{s_j}^1, \rho_{s_j}^1), (x_{s_j}^{1'}, 1-\rho_{s_j}^1); (x_{s_j}^2, \rho_{s_j}^2), (x_{s_j}^{2'}, 1-\rho_{s_j}^2); \cdots; (x_{s_j}^k, \rho_{s_j}^k), (x_{s_j}^{k'}, 1-\rho_{s_j}^k) \rangle$ 表示,且有 $x_{s_j}^i[i] < x_{s_j}^{i'}[i]$。对任意节点:若存在节点有 $x_{s_j}^i[i] > l_i$,则该节点为异常的节点,组合系统不满足资源约束。若对所有节点 $x_{s_j}^{i'}[i] \leqslant l_i$,则节点为正常的节点,组合系统满足资源约束。若存在一点有 $x_{s_j}^{i'}[i] > l_i \geqslant x_{s_j}^i[i]$,则该节点为正常节点的概率为 $\rho_{s_j}^i$,且该节点为不稳定节点。若系统存在 m 个不稳定节点,则系统受到资源约束的概率为 $1 - \prod_{j=0}^{m-1} \rho_{s_j}^i$。

如图3所示某随机性资源接口网络构造的可达图 M,有 S_0, $S_1, \cdots, S_7$ 共8个节点,每个节点标注了 ε_1、ε_2 资源使用的信息。当给定的资源约束 $L = \langle 9;10 \rangle$ 时,可判断出该可达图所有节点满足资源约束条件,则该组合系统满足资源约束。当给定的资源约束 $L = \langle 5;6 \rangle$ 时,节点 S_2 对于 ε_2 资源的使用是不满足资源约束的,对于 ε_1 有0.6的概率是满足资源约束,0.4的概率是不满足资源约束的;对于节点 S_4 不满足资源约束条件。

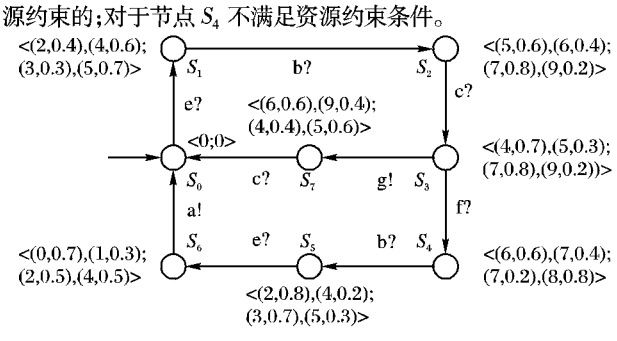


图3 可达图 M

基于上述讨论,针对文献[5]所提出的算法进行改进以适用于随机性资源接口自动机。算法的输入为可达图及资源约束 $L=\langle l_1,l_2,\cdots,l_K\rangle$,输出结果为组合系统是否满足资源约束;在不满足资源约束的情况下,输出为属于不满足类型还是不稳定类型并且输出相应的节点集合。在算法 1 中,遍历路径存在 *current_path* 中,L 存放长度为 K 的资源约束;使用 $X(node)$ 表示节点的资源使用信息,其中 $X_1[i]$、$X_2[i]$ 分别为第 i 种资源在不同环境下的使用信息,且 $X_1[i] < X_2[i]$;*abnormal* 存放不满足资源约束的节点,*instability* 存放不稳定节点,*satisfied* 是一个布尔变量,当不存在异常节点时为 true,否则为 false。

算法 1　检查组合系统是否满足资源约束算法。

```
current_path: ={S0}; L=⟨l1, l2, …, lk⟩;
X:=⟨0, 0, …, 0⟩; abnormal:=∅; instability:=∅; satified:=true;
repeat
  node:=取 current_path 中最后一个节点
  if node 的后继节点均已访问 then 删除 current_path 中的最后一
    个节点;
  else begin node:=取 node 的一个未访问过的后继节点;
  X:=X(node);
  for i=1 to K
    if X1[i] >li then satisfied:=false;
    {
      if satisfied:=false then 将 node 加入到 instability 集中;
        将 node 加入到 current_path 中;
    }
    if X1[i] ≤li <X2[i] then satisfied:=false;
    {
      if satisfied:=false then 将 node 加入到 instability 集中;
        将 node 加入到 current_path;
    }
  end
until current_path ={};
if abnormal:=∅ && instability:=∅ then return true;
if abnormal:=∅ && instability:! =∅
  then return astable else return false;
```

算法 1 所描述的为组合系统在变化的环境中是否满足资源约束的验证问题,与之相对应的是一个组合系统能够正常运行所需要最小资源的问题。具体见算法 2。

算法 2　组合系统运行资源最小量算法。

```
current_path:={S0};
L:=⟨0, 0, …, 0⟩; X=⟨0, 0, …, 0⟩;
repeat
  node:=取 current_path 中最后一个节点;
  if node 的后继节点都已访问 then 删除 current_path 中最后一个
    节点;
  else begin node:=取 node 的一个未访问过的后继节点;
    X:=X(node);
    for i=1 to K;
      if X2[i] >li then bi:=X2[i];
    将 node 加入到 current_path 中;
  end
until current_path ={};
```

3　实例应用

因网络带宽不确定,网上书店系统可分为文本版和标准版,所以网上书店系统[6]对带宽资源的使用具有随机性。使用随机性资源接口自动机对网上书店系统带宽资源使用情况进行描述。图 4(a)和(b)分别表示演化后的书店系统连接件的角色 seller、构件 Publisher 的端口 publish 所对应的随机性资源接口自动机描述。

针对图 4,当给定的带宽资源为 1 MBps 时,即带宽资源约束为 4 个单位,系统在任何状态下都不会受到资源约束,则用户使用任何版本的系统都不受带宽资源的约束。当给定的带宽资源为 768 kBps 时,即带宽资源约束为 3 个单位,系统连接件的角色 seller 在状态 2、状态 4 以及构件 Publisher 的端口 publish 的状态 1 受到资源约束的概率分别为(0.5,0.6,0.6),则系统受到资源约束的概率为 $1-0.5\times0.4\times0.4=0.92$,因此用户只能使用文本版的书店系统。当给定带宽资源为 512 kBps 时,即带宽资源约束为 2 个单位,系统连接件的角色 seller 在状态 2 受到资源约束,那么用户不能使用网上书店系统的任何版本。

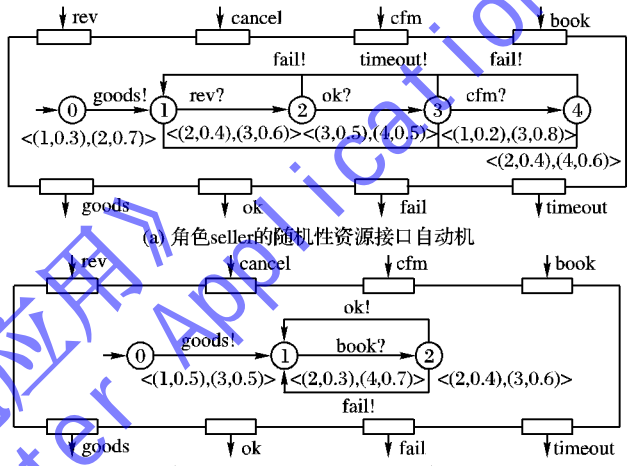


图 4　网上书店系统随机性资源接口自动机模型

4　模型验证

本文采用模型检验工具 Spin[9-10] 对演化后的书店系统角色 seller 模型进行验证,通过对状态 2 的随机性资源满足性的验证以说明模型的正确性。在应用 Spin 验证模型前,首先要使用 Promela 语言建模,根据文献[11]给出的从接口自动机模型到 Spin 的系统模型描述语言 Promela 的转换规则,将图 4 的模型转换为如下的 Promela 模型。模型中 k 表示给定带宽资源最小单位的数目。由于篇幅有限,下面给出部分代码。

```
mtype = { good, rev, ok, fail, cfm, cancel, Timeout}
proctype Publisher( chan ch1, ch2, ch3, ch4, ch5, ch6)
{ int k =
    do: :
    sv0:
    do: :
      ch! goods;
      goto sv1;
    od;
    sv1:
    do: :
    if
      : : (k > =3) -> ch2?rev;
      goto sv2;
      fi;
    od;
    sv2:
    if
      : : ch3! fail -> goto sv1;
```

```
        ::ch3! ok -> goto sv3;
        fi;
    sv3:
    if
        ::ch5?cancel -> goto sv1;
        ::ch5?cfm -> goto sv4;
        ::ch5! timeout -> goto sv1;
        fi;
    sv4:
    if
        ::ch6! ok -> goto sv1;
        ::ch6! fail -> goto sv1;
        fi;
    od
}
```

对于 Promela 所描述的模型，利用 Spin 分别在 $k=4$ 以及 $k=2$ 的情况下验证系统的可达性。验证结果如图 5(a)、5(b)所示。图 5(a)表示在 $k=4$ 情况下，系统的可达路径数目为 29；图 5(b)表示在 $k=2$ 的情况下可达路径数目为 5，这是因为状态 2 不满足资源约束，系统无法到达状态 2，从而导致可达路径少于 $k=4$ 的可达路径。反映实际系统在具有随机性的环境中所能够提供给用户的服务是不同的。所以，检测结果表明了模型的正确性。

(a) k=4的验证结果

(b) k=2的验证结果

图 5 Spin 验证结果

5 结语

本文工作着眼于具有不确定性资源的因特网环境下网构软件对环境自适应性的可信度研究，扩展了接口自动机为随机性资源接口自动机，应用扩展后的随机性资源接口自动机对网构软件系统资源进行形式化描述；给出了资源约束满足和最小资源判断算法，通过对构件组合系统的行为是否满足资源约束进行判断，从而确定网构软件系统对具有随机资源的环境适应性的可信性。下一步工作是开发相应的自动分析和验证工具的原型。

参考文献：

[1] 杨芙清，梅宏，吕建，等. 浅论软件技术发展[J]. 电子学报，2003，26(9)：1104－1115.

[2] 吕建，马晓星，陶先平，等. 网构软件的研究与进展[J]. 中国科学 E 辑：信息科学，2006，36(10)：1037－1080.

[3] 曾红卫，缪淮扣. 构件式系统的建模与验证[J]. 计算机科学与探索，2008，2(2)：198－205.

[4] 霍健，杨秋松，肖俊超，等. 一种形式化的组合化软件过程建模方法[J]. 软件学报，2011，22(1)：1－16.

[5] 胡军，黄志球，曹东，等. 网构软件的资源自适应性的形式化分析与验证[J]. 软件学报，2008，19(5)：1186－1200.

[6] 包书勇，王忠群. 网构软件演化的业务一致性验证方法[J]. 计算机工程，2011，37(17)：29－31.

[7] DE ALFARO L, HENZINGER T A. Interface automata[C]// Proceedings of the 8th European Software Engineering Conference and the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering. New York: ACM, 2001: 109－120.

[8] 桑海，张明清，唐俊. 基于可达图的仿真组件接口设计一致性验证[J]. 计算机仿真，2010，27(4)：75－79.

[9] BEN-ARI M. Principles of the spin model checking[M]. London: Springer-Verlag, 2008.

[10] HOLZMANN J G. Spin model checker: primer and reference manual [M]. Boston: Addison-Wesley, 2003.

[11] 谭亮，曾红卫. 基于接口自动机的 Web 应用验证[J]. 计算机工程与应用，2009，45(3)：70－73.

(上接第 3043 页)

表 3 本文算法与其他经典启发式算法的计算速度与参数个数

问题集	Solomon[1]	Potvin 等[4]	Ioannou 等[5]	本文算法
C1	25.30	856.90	168.00	1.23
C2	43.00	1042.70	294.00	1.45
R1	24.70	882.20	141.00	1.84
R2	62.60	1884.50	384.00	1.43
RC1	23.80	891.60	126.00	1.65
RC2	51.70	1428.70	300.00	1.43
平均	38.52	1164.43	235.50	1.51

3 结语

求解 VRPHTW 的插入启发式算法是一类重要的求解 VRPHTW 启发式算法，它不仅有速度快、参数设置少的优点，而且也是智能启发式算法的重要组成部分。本文提出了可广泛运用于各类求解 VRPHTW 启发式算法的时差插入检测法，并详细介绍了该算法的启发规则、算法构架。最后运用仿真程序对时差插入启发式算法进行测试，得出了该算法的最佳参数。与三种经典插入启发式算法的比较结果表明：本文算法求解质量优于 Solomon 的算法与 Potvin 的算法，但比 Ioannou 的算法要差一些。但总体分析认为，本文算法参数要求少，结构简单，求解质量较高，能在较快的时间内找到满意解。

参考文献：

[1] BRAYSY O. Vehicle routing problem with time windows, Part Ⅱ: metaheuristics[J]. Transportation Science, 2005, 39(1): 119－139.

[2] SOLOMON M M. On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints[J]. Networks, 1986, 16(2): 161 － 174.

[3] SOLOMON M M. Algorithms for the vehicle routing and scheduling problems with time window constraints[J]. Operations Research, 1987, 35(2): 254－265.

[4] POTVIN J-Y, ROUSSEAU J-M. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows[J]. European Journal Operation Research, 1993, 66(3): 331 － 340.

[5] IOANNOU G, KRITIKOS M, PRASTACOS G. A greedy look-ahead heuristic for the vehicle routing problem with time windows[J]. Journal Operation Research Society. 2001, 52(5): 523－537.

[6] 潘立军，符卓. 求解带时间窗车辆路径问题的插入检测法[J]. 系统工程理论与实践，2012，32(2)：319－322.